Project Report for Data-pipeline

Introduction

Often DSTI students must understand the way data flows from websites and manipulate it. Under the course of Data Pipeline, an attempt was made to create a dummy tourism website and control its data.

As stated above, this project is created for a sample tourism website or marketplace that enables small travelling companies in India to sell their respective tickets. The intention is that the data analyst should be able to access data that may affect the company's business.

Data Structure

To create such data, one can consider multiple database formats: XML, SQL or JSON or any other. However, most of the time, a significant comparison is made between semi-structured databases (XML) vs structured database (SQL) formats.

Why use XML?

Dealing with NULL values.

Suppose the expected type of data we may encounter in such projects is considered (schematically shown in figure 1, on page 3). In that case, it is easy to understand that not all trips will have the same number of customers or not all trips will have the same number of activities. If SQL data is used in such a scenario, it will create many NULL values for non-existent fields.

While querying such data, the user must take care of such values each time. Also, it is not possible to get rid of all NULL values in some scenarios. On the other hand, XML can handle such data without any issue.

Dealing with memory and processing time

Since the number of entries in XML are lesser than that created in the SQL files, thanks to the fact that NULL values need not exist in XML databases, memory utilised by the XML database is lesser. Also, consequently, the processing time required for XML data is lesser.

Flexibility

SQL can have superior structural integrity but has a rigid structure. On the other hand, using XML, a programmer can create any variety of elements and attributes.

So, in the end, in the case of web applications, space, processing time, and flexibility are critical commodities. Hence, it is natural to use XML format in this project.

Software Requirements

The software package used for the project is Notepad++ with relevant extensions to process XML and XSL transformations. Also, the 1.0 XML version was used throughout the projects to define XML database, XML Schema and XSL transformations.

XML file structure

As shown in Figure 1 (on the next page), the root element is Tourism. The main five elements are Trip, Company, Customer, Destination, and Activity.

Trip

Functionally, “Trip” is one of the primary elements and contains information about the trips sold on the website. Then, each trip has a

- title
- start date
- company
- customers
- destination
- activities
- remarks

Ideally, such information related to a trip is collected from the companies which are ready to sell their trips on the ‘Tourism’ website that we are building.

Also, each trip has attributes that state the duration, cost and genre of the trip. Under element title, start dates

Company

There are a total of three companies that relate to the website. Each Company is uniquely connected with each trip with the help of key and keyrefs. Also, a web address of the companies and their contacts are provided in case the prospective customers wish to contact them back.

Customer

This element contains the first name, last name, and emails of all the customers who have already booked their trips using the ‘Tourism’ website. This is not the company data.

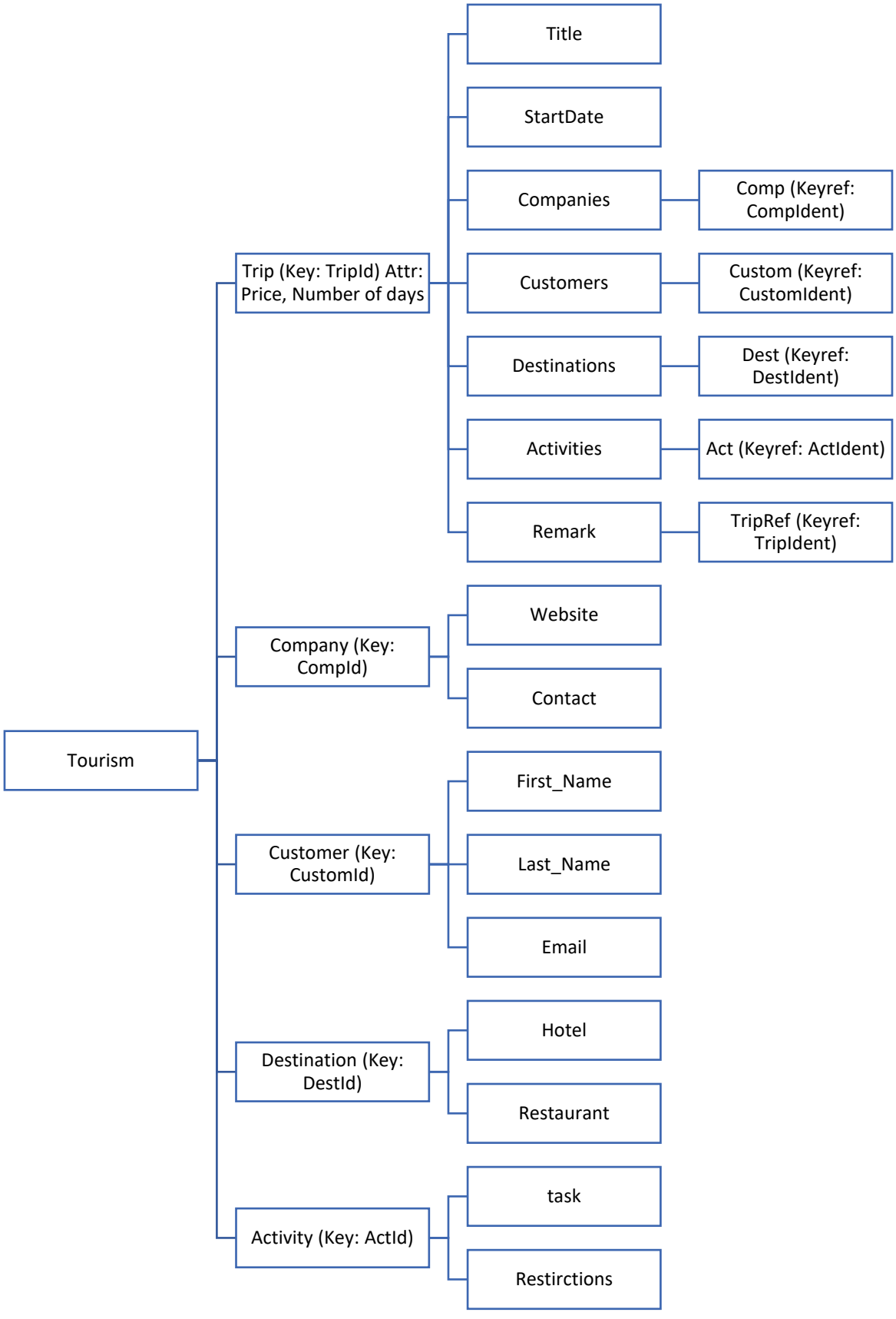

Figure 1 Tree-view of the XML file

Destination

The destination element contains information about the local restaurants and hotels. Such information is collected from different sources and not exclusively through companies selling their services.

Activity

Under this element, different activities provided by the organisers or companies are provided. Each activity has a task and set of restrictions. In case the guests must pay extra charges for an activity, they can refer to these restrictions.

XSL Transformations

A total of 4 scenarios were envisioned for the data transformation. Three were to create HTML output, whereas the last one was to create output in XML format.

Scenario 1: All trips with vital parameters arranged in descending order of price

Usually, many people like to know their options for trips and compare their budgets instantaneously. So, in this case, data is manipulated such that the prospective client will be able to compare each trip arranged by the descending price of a trip in an HTML file. Also, the following vital parameters of the trip are considered as follows.

- Title
- StartDate
- Price
- Number of Days
- Genre
- Destination

Scenario 2: All trips to Himalaya

Also, many people like to visit some popular destinations such as Himalaya in this case. So, the XSL transformation offers similar data to that given in the previous scenario, but only for the destination called Himalaya.

Scenario 3: Short trips

Often, travellers may not have long vacations, so they may search for all those trips that can be completed within ten days. So, in such a scenario with a conditional, all trips with all vital parameters compared for the trips with a duration of 10 days or less.

Scenario 4: Knowing all activities

As a business need within the company, if data regarding all activities in all trips must be transferred to another team or department, one needs to extract only that data from the main XML file. So, in such a scenario, an XML file is created that contains only activities.

Future Scope

Add more specific data

By adding more namespaces that can deal with weblinks, one can add specific weblinks to book a trip. Also, icons, pictures and geographical coordinates can be added to the database. Such addition is always more pleasing to the eye and helps customers make decisions quickly.

Create better interlinking

The data can be interlinked in an even better way so that querying can be much more complex to replicate searches on modern interfaces.